FREE—ALGORITHM FOR SOLUTION OF AN SLBM MULTIPLE CONSTRAINT MISSION PROBLEM

→ carried out in FCS

Ms. Shawna M. Davis and Mr. Davis L. Owen

The TRIDENT missile consists of four stages and multiple reentry bodies (RBs). Each RB flies a trajectory that is subject to its own constraints as well as being constrained by the trajectories of other reentry bodies. The ensemble of trajectories of the individual reentry bodies needs to be designed to avoid all constraints. Each RB trajectory must stay within the RB survival envelope. The ensemble of trajectories must minimize energy requirements for the missile, allow spatial separation of the trajectories, and control the time/impact patterns to maximize effectiveness.

Currently, each Trident missile is programmed to allow all its RBs to reach their respective targets without encountering intramissile constraints. However, the current process removes violations successively and can result in a solution that is suboptimum. This article describes a new algorithm where a system of constraint equations is developed to yield simultaneous solutions for all RB trajectories. These equations are solved by a least-squares method and result in a solution that is superior to the existing methodology. When developed fully, this algorithm has the potential to solve the constraint problem while not degrading system performance.

INTRODUCTION

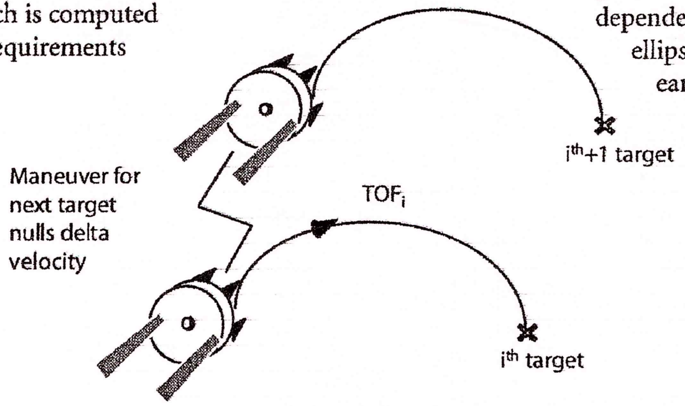
Each RB carried by the TRIDENT missile is given an independent mission and flight time. The flight times are controlled by the time-of-flight (TOF) preset assigned to each RB. In the absence of constraints on any RB path or between any pair of RB trajectories, the TOFs are assigned to minimize the energy used by the missile to deploy the RBs. If constraints are violated, the TOFs are used to resolve constraints.[1]

The three major constraints affecting TOFs are to:
- Guarantee separation or spacing between RBs during flight
- Assure all RBs are within the RB survival envelope
- Avoid undesirable time and geometric relationships between RBs at fuzing

The first two constraints are hard constraints and are enforced by rigorous mathematical algorithms. The third constraint is handled indirectly by modifying the mission and by observing the effect on the TOF computations. Unlike the first two constraints, it is acceptable to minimize the occurrences of this constraint.

CURRENT ALGORITHM

All RBs are numbered and referenced according to the order that they will be released. The TOFs are currently computed recursively based on the RB release order. This process is initiated by the computation of the first RB's TOF, which is computed to satisfy mission requirements independent of any of the issues contained in this article. All TOF computations, including first body TOF, are dependent upon ship's position, and results can vary significantly over the projected patrol area.



Min energy TOF minimizes delta velocity

Figure 1—Minimum Energy TOF

The first two constraints are currently resolved directly in the computation of the TOFs. The conflict resolution process is begun by requiring that the initial estimate of each RB's TOF minimizes the velocity increment required to place it on its desired trajectory. This is denoted as the minimum energy (min energy) TOF (see Figure 1). The algorithm enforces the spacing constraint by computing time stayout windows for each RB based on the trajectories for all previously processed RBs (see Figure 2). For example, if the n^{th} TOF is being computed, time stayout windows are computed with respect to the first through n^{th}-1 RB trajectories. If the min energy TOF falls into one of these stayout windows, TOF is increased to the minimum value outside of any stayout windows. If the final value of TOF violates the RB survival envelope, the algorithm then must search for an acceptable value. The RB survival envelope determines the minimum and maximum values of TOF.

The third constraint on TOF is due to impact time/position geometry and is expressed as a series of time-dependent ellipses. These ellipses are on the earth's surface near the impact area and, for each pair of RBs, can be written as a function of their TOF differences. The ellipses are used to determine an Impact Effectiveness (IE) ratio for each RB pair (see Figure 3). A ratio of 1.0 or greater is acceptable. Impact constraint violations can normally be minimized or eliminated by modifying the mission assigned to the missile in a way that changes the TOFs. Common techniques include interchanging the targets assigned to the RBs or iterating the spacing parameters.

Two disadvantages of the present methodology prompted this investigation into an improved algorithm. First, the impact geometry constraint is ignored during the actual TOF

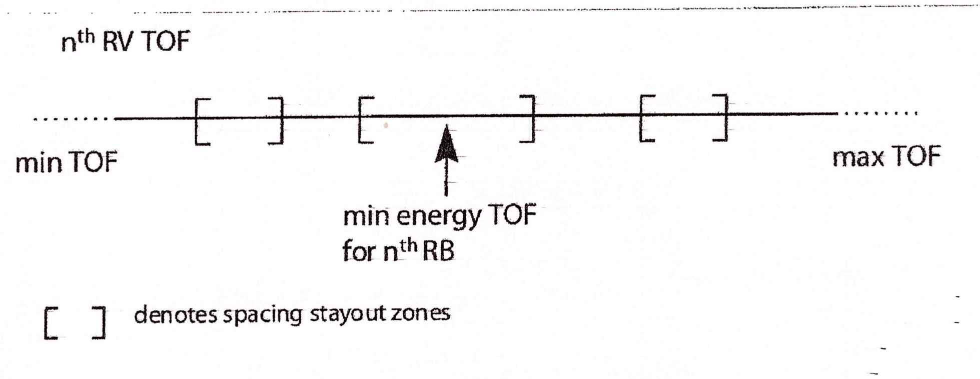


Figure 2—Window Concept for Spacing

construction and is satisfied later by a process of modifying the mission and observing the result. Second, it has been observed that the rule to increase TOF to satisfy spacing sometimes degrades the final solution. For these reasons, the present methodology frequently gives a solution that is suboptimum.

The new algorithm described in this article unifies all TOF constraints into one solution. The goal is to find the set of TOFs closest (least squares) to min energy that satisfy all constraints. This goal preserves the best features of the present methodology while repairing its deficiencies. First, the min energy concept is preserved and strengthened. Second, the technique of removing TOF constraints by time windows is preserved. The increase/decrease TOF issue for spacing is resolved by allowing the least squares to pick the best solution. Previous attempts to solve this problem with time windows floundered due to the fact that the spacing stayout windows are in RB release order, but the impact time windows are in impact order. This dilemma was resolved by replacing the recursive computation of TOFs by a simultaneous solution for all TOFs.

FREE ALGORITHM

There are two types of spacing constraints. The first constraint is actually a minimum velocity between consecutive RB deployments, which is mathematically equivalent to a variable spacing. The second constraint ensures spacing between all consecutive and nonconsecutive pairs. Since the spacing for nonconsecutive pairs is normally used to maximize IE, it became extraneous for FREE and was therefore eliminated. FREE recognizes the remaining spacing constraints on consecutive pairs and expresses them as time stayout windows.

RB survivability constraints are currently not defined in terms of a time window. The conditions the RB would encounter at reentry are predicted as a function of TOF and tested against the RB reentry envelope. This poses a slight problem for the FREE algorithm since a violation of minimum or maximum TOF is known only after a solution is found. However, this is not a serious concern since avoiding survivability constraints is one of the factors in the computation of the first RB TOF.

With FREE, the impact geometry constraint is used in conjunction with the spacing and survivability constraints. A couple of simple, but important, changes were made to the impact geometry computations. Originally, an IE ratio for each pair of RBs would be computed based on their impact time. The new method assumes all TOFs are initially the same. It then varies the TOF of one RB of each pair. This variance is done in increments from $-y$ to $+y$, where y is a given delta time from the TOF. With each TOF adjustment, the IE ratio is computed. The result is that for each pair of RBs, a time window where the IE ratio is above

1.0 can be found. Impact time windows are no longer dependent upon the impact order but can remain numbered based on their release order. This makes the IE time windows, or stayin windows, compatible with the other constraint windows.

All time windows are determined for each specified launch point in a given patrol area. An algorithm was developed to take the time windows for each launch point and determine one set of time windows that encompass the patrol area. An example of how these time windows are combined is provided in Figure 4. In Figure 4(a) the combining of stayin windows is depicted. Figure 4(b) depicts the combining of stayout windows. Once the time windows have been defined for the launch area, another algorithm is needed to combine the stayin and stayout windows for each pair. The stayout windows apply only to consecutive RBs. Stayin windows apply to both consecutive and nonconsecutive pairs (see Figure 5). The result will be stayin windows for each pair of RBs that encompass the entire patrol area. Any TOF within these windows is valid.

There are many benefits to describing the constraints as time windows. First, there is no need to iterate to find a conflict-free TOF. The constraints can be solved simultaneously. Second, there is no restriction on the direction of adjusting TOF. It can be increased or decreased from min energy TOF. Third, the optimum TOF can be found by forcing the TOF to be as close to min energy TOF as possible.

OPTIMIZATION ALGORITHM

The optimum TOF is solved for using a least-squares algorithm. The algorithm implemented takes the form of a least-distance programming technique (LDP),[2] which solves the system of equations of the form

$$\text{minimize } \|x\| \text{ subject to } Gx \geq h \qquad (1)$$

where

x = Solution vector of size n-1, where n equals the number of RBs to release. This vector contains the time increment from min energy TOF for each of the RBs, except the first.

G = Matrix of size $m \times n$-1 that contains ones, negative ones, and zeros, where m equals $(n\text{-}1)n$. This matrix is used to reference the time increments (x-components) to their corresponding stayin windows (h-components).

h = Constraint vector of size m that contains the start and ending times of the windows.

Gx is greater than or equal to h if every element of Gx is greater than or equal to the corresponding element of h.

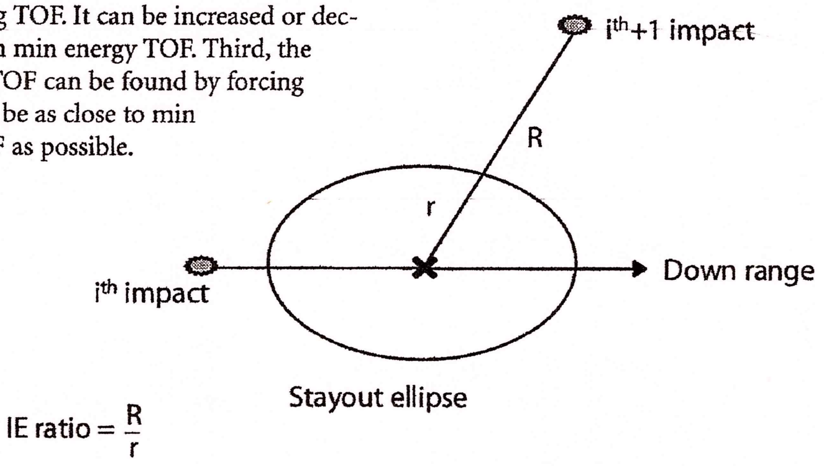
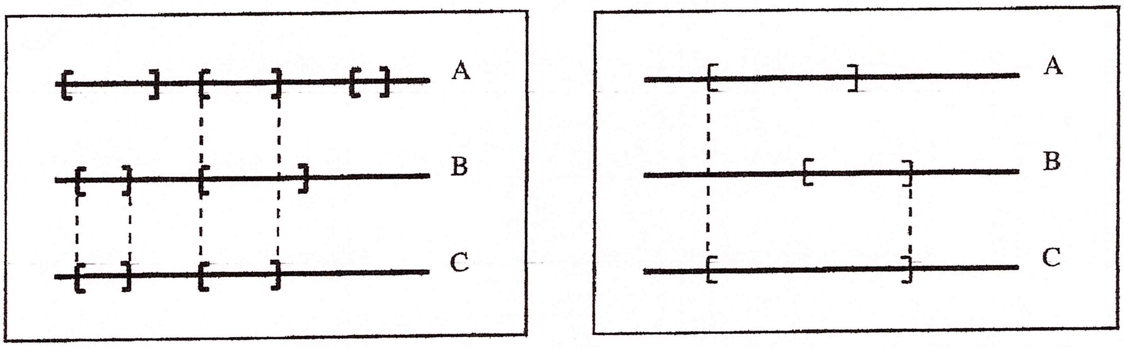


Figure 3—Impact Effectiveness Ratio



(a) Stayin Windows - IE constraints

(b) Stayout Windows - Spacing constraints

Figure 4—Combining of Windows for All Launch Points in Given Patrol Area

A: one through n^{th} launch points B: n^{th}+1 launch point C: one through n^{th}+1 launch points

The LDP technique described by Reference 2 obtains a solution to an arbitrary set of linear inequalities $Gx \geq h$ by reducing the problem to a form where the nonnegative least-squares method (NNLS)[2] can be invoked. NNLS computes x for the following least-squares problem

$$\text{minimize } \|Ax - b\| \text{ subject to } x \geq 0 \qquad (2)$$

This is closed-form and usually terminates after a reasonably small number of iterations. If $Gx \geq h$ has a solution, LDP will compute x of minimal norm, satisfying the inequalities. If the constraints are not compatible, the algorithm indicates that fact and then terminates.

As illustrated in Figure 5, it is possible to obtain more than one stayin window for any particular pair of RBs. The least-squares method is restricted to using one stayin window per pair. It is not feasible to check every combination of windows through the least squares to determine the best solution. However, it was found that almost any combination that produced a valid solution was as good or better than the current algorithm.

APPLICATIONS

FREE can be utilized in various ways. Its original design purpose was to deconflict a mission over an entire patrol area. The only

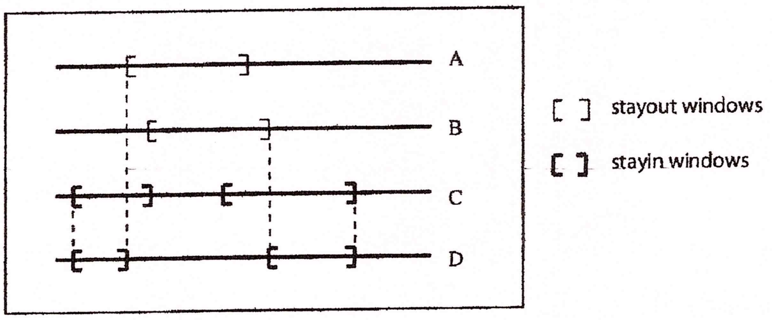


Figure 5—Combining of Windows for Consecutive Pairs of RBs

A: 1st spacing constraint B: 2nd spacing constraint C: impact effectiveness D: final resulting stayin windows necessary input is the minimum and maximum latitudes and longitudes, and the degree of increment. If it exists, FREE will find one TOF for each RB that avoids spacing restrictions, maintains reentry survivability, maximizes IE, and minimizes the use of propellant. It is a simple, one-step process to deconflict the patrol area.

Often, it is desirable to know how the solution for the patrol area will work just outside the patrol area boundary. FREE has the capability of finding a solution for the given patrol area and then checking that solution over a specified expanded area.

Another method for using FREE is to deconflict strictly one launch point. FREE is extremely powerful in this point-by-point mode. It is useful when trying to find a solution for a difficult launch point. FREE will attempt to resolve the constraints for the given launch point. If it is not physically possible to resolve the constraints, the output from FREE will explain the reason: reentry survivability, spacing, IE, propellant usage, or a result of the interaction of these constraints.

In some situations, the mission may not be resolvable. If targets are tightly packed, it may not be possible to obtain an IE ratio above 1.0. If this target package cannot be modified, then it is still possible to use it, albeit with degraded effect. This degradation can be minimized by allowing IE ratios slightly below 1.0. The current conflict resolution process does not allow any flexibility in the IE ratio limit. However, the ratio limit in FREE is an adjustable variable. This allows FREE to minimize the negative effects of unavoidable conflicts.

RESULTS

The two main uses of FREE would be to deconflict a single launch point and to deconflict a patrol area. Many test cases were used as a checkout for FREE. In all scenarios, FREE did as well or better than the current process. Figures 6 and 7 are used to illustrate the improvement obtained by using FREE. All the examples are fictitious and do not indicate latitudes or longitudes.

The first example (Figure 6) illustrates the usefulness of FREE in the point-by-point mode. Each small square represents a launch point, of which there are approximately 1,000. The white area indicates launch points that were deconflicted. The light gray rectangle in the middle is an unachievable region that contains the targets. Dark gray areas indicate launch points that cannot be deconflicted. There are no TOFs that

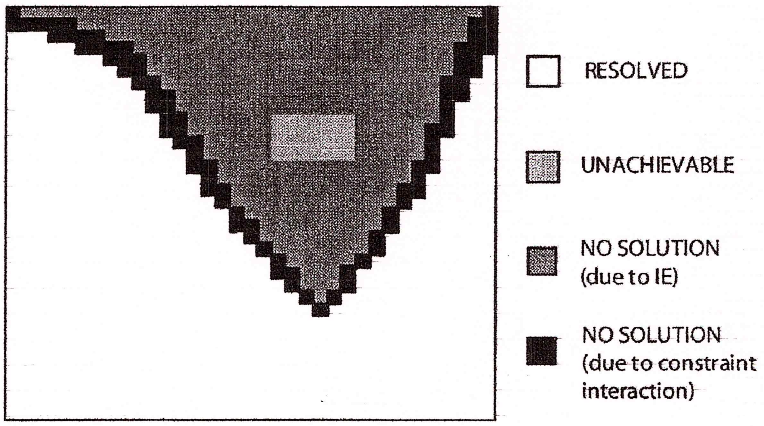
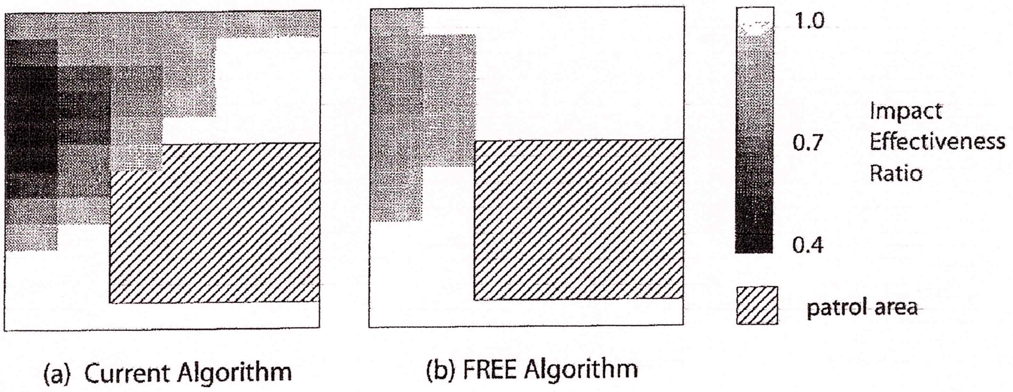


Figure 6—FREE Results Using Point-By-Point Method



(a) Current Algorithm (b) FREE Algorithm

Figure 7—Impact Effectiveness Ratio Results for Deconflicting a Patrol Area

give an IE ratio above 1.0. The only way to use the mission for these launch points is to allow a ratio below 1.0. Black areas indicate launch points with no solution. Interaction of the spacing constraint with the IE constraint for a consecutive pair of RBs precludes a solution. A valid solution with an IE ratio above 1.0 may be possible if the release order is modified so that the conflicting pair is nonconsecutive instead of consecutive.

The second example (Figure 7) shows how FREE can be used to deconflict an entire patrol area. For the current process, the first step was to run a program to remove spacing and survivability conflicts within the patrol area. Next, another program was run to analyze the IE ratio. In this example, the entire patrol area and expanded area had an IE ratio of about 0.4. The final step was to run another program to improve the ratio over the patrol area. Figure 7(a) shows the results of the IE ratio. The outer box indicates the expanded area, and the cross-hatched box inside the expanded area indicates the patrol area. The current process deconflicted 96 percent of the patrol area and just under 50 percent of the expanded area. In contrast to this, the patrol area could be deconflicted by one simple execution of FREE. Figure 7(b) shows the result from FREE. All spacing and survivability constraints were avoided for the entire area. The IE ratio is above 1.0 for 100 percent of the patrol area and 75 percent of the expanded area.

CONCLUSIONS

Controlling flight time to avoid constraints is a topic of current interest in the Submarine Launched Ballistic Missile (SLBM) community. FREE is the initial prototype algorithm developed in response to these concerns. The ideas contained in FREE should represent the foundation for future constraint resolution algorithms. It is easily used and can be adapted for many applications. FREE is a good example of how existing concepts and methodology can be reexamined and reassembled into powerful new algorithms.

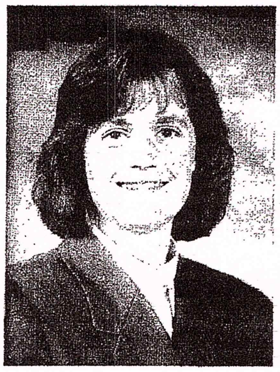
REFERENCES

1. Ray, J.F., *Time of Flight Generation for Constrained Ballistic Trajectories*, NWL TR-2955, Naval Weapons Laboratory, Dahlgren, VA, May 1973.

2. Lawson, C.L. and Hanson, R.J., *Solving Least Squares Problems*, Prentice-Hall, New Jersey, 1974.

The Authors

Ms. Shawna M. Davis



Ms. Shawna M. Davis earned a B.S. with honors in aerospace engineering from the University of Cincinnati in 1994. Since arriving at the Naval Surface Warfare Center, Dahlgren Division (NSWCDD) in 1991 as a Cooperative Education Student, she has worked primarily on six degree-of-freedom missile simulations. She has identified and developed innovative trajectory modeling methodologies and applied them in the investigation of advanced system concepts and SLBM weapon system evaluation. Ms. Davis has coauthored an AIAA paper on high-angle-of-attack aerodynamics. In 1994, the Chief of Naval Operations recognized her as the Navy's Co-op of the Year in Science and Engineering. Ms. Davis is currently completing requirements toward an M.S. degree in aerospace engineering from the University of Colorado through NSWCDD's Academic Fellowship Program. She is a member of Sigma Gamma Tau and Tau Beta Pi honor societies, and the American Institute of Aeronautics and Astronautics.

Mr. Davis L. Owen

Mr. Davis L. Owen received a B.S. degree in mathematics from Millsaps College in May 1964 and an M.S. in mathematics from the University of Southwestern Louisiana in June 1966. Since then, he has been employed at NSWCDD, where he has been involved with various aspects of SLBM presetting and performance. His work has centered around trajectory simulation including gravity, reentry, and fuzing. In November 1986, Mr. Owen received the Navy Meritorious Civilian Award for his work on gravity modeling.